

Задача А. Решето Эратосфена

Имя входного файла: `sieve.in`
Имя выходного файла: `sieve.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

По введенным числам A и B вывести все простые числа в интервале от A до B включительно.

Формат входных данных

В единственной строке вводятся два числа $1 \leq A \leq B \leq 100000$

Формат выходных данных

Вывести в одну строку все простые числа в интервале от A до B включительно

Примеры

sieve.in	sieve.out
2 2	2
1 100	2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Задача В. Обратное по модулю

Имя входного файла: `inverse2.in`
Имя выходного файла: `inverse2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны два целых числа — a, m ($0 \leq a < m$). Нужно найти такое целое x , что $a \cdot x \equiv 1 \pmod{m}$.

Формат входных данных

На первой строке два целых числа — a, m ($0 \leq a < m \leq 10^{18}$).

Формат выходных данных

Если такого x не существует, выведите -1 . Иначе выведите целое x ($0 \leq x < m$). Если ответов несколько, выведите любой.

Примеры

<code>inverse2.in</code>	<code>inverse2.out</code>
7 30	13

Задача С. Китайская теорема

Имя входного файла: `chine.in`
Имя выходного файла: `chine.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Решите в целых числах систему уравнений

$$\begin{cases} x \equiv a \pmod{n} \\ x \equiv b \pmod{m}, \end{cases}$$

где n и m взаимно просты. Среди решений следует выбрать наименьшее неотрицательное число.

Формат входных данных

Входной файл содержит четыре целых числа a, b, n и m ($1 \leq n, m \leq 10^6, 0 \leq a < n, 0 \leq b < m$).

Формат выходных данных

В выходной файл выведите искомое наименьшее неотрицательное число x .

Примеры

<code>chine.in</code>	<code>chine.out</code>
1 0 2 3	3
3 2 5 9	38

Задача D. Система линейных сравнений

Имя входного файла: `chinese.in`
Имя выходного файла: `chinese.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дана система из двух линейных сравнений:

$$\begin{cases} x \equiv a \pmod{n}, \\ x \equiv b \pmod{m}; \end{cases}$$

где числа n и m не обязательно взаимно простые. Решите эту систему или определите, что она не имеет решений.

Формат входных данных

В первой строке входного файла записано единственное число $1 \leq t \leq 100\,000$. В следующих t строках содержатся по четыре целых числа a, b, n, m , задающих одну систему сравнений. Все числа не превосходят по модулю 10^4 , $n > 1$, $m > 1$.

Формат выходных данных

Программа должна вывести t строк, по одной на каждую систему.

В случае, если система не имеет решений, выведите строку "NO".

В случае, если решение есть, то необходимо вывести слово "YES" и два таких числа x_0 и p , $0 \leq x < p$, такие, что множество чисел $x = x_0 + kp$, где k — произвольное целое число является решением данной системы.

Примеры

chinese.in	chinese.out
3	YES 38 45
3 2 5 9	YES 1 45
1 1 5 9	NO
7 13 20 24	

Задача Е. Проверка на простоту

Имя входного файла: `prime.in`
Имя выходного файла: `prime.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дано натуральное число N . Определите, является ли оно простым.

Формат входных данных

Программа получает на вход одно целое число N , $2 \leq N \leq 10^{18}$.

Формат выходных данных

Если число N простое, программа должна вывести YES, для составного числа программа должна вывести NO.

Примеры

<code>prime.in</code>	<code>prime.out</code>
3	YES
4	NO

Задача F. Больше простых!

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 15 секунд
Ограничение по памяти: 64 мегабайта

Найдите все простые числа не большие n . Поскольку n в этой задаче не просто большое, а прямо здоровенное, для того чтобы проверить, что вы нашли числа правильно, мы попросим вас посчитать от найденных чисел специальный хеш.

Хеш будет считаться по следующему алгоритму. В начале переменная $h = 0$. После каждого очередного встреченного простого числа p_i , будем пересчитывать h по формуле $h = h \cdot x + p_i$, при этом будем игнорировать переполнение знакового 32-битного целого типа. Значение переменной h в конце — это хеш, который вам нужно вывести.

Формат входных данных

Входной файл содержит два числа n ($2 \leq n \leq 10^9$) и x ($1 \leq x \leq 10^9$).

Формат выходных данных

Выведите полученный хеш.

Примеры

стандартный ввод	стандартный вывод
10 10	2357
11 100	203050711
1000000000 2	1576840463

Задача G. Циклический шифр

Имя входного файла: `circular-cipher.in`
Имя выходного файла: `circular-cipher.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Эта задача взята с одного из раундов *codeforces.com*. Выражаем благодарность координатору Codeforces Глебу **GlebsHP** Евстропову, а также Михаилу **MikeMirzayanov** Мирзаянову за системы *Polygon* и *Codeforces*.

Вам задан набор из n последовательностей. Каждая из последовательностей состоит из целых положительных чисел, не превосходящих m . Все числа внутри одной последовательности различны, но одно и то же число может встречаться в разных последовательностях. Длина i -й последовательности равна k_i .

Раз в секунду числа в каждой последовательности циклически сдвигаются на одну позицию влево, то есть числа на позициях $i > 1$ переходят на позиции $i - 1$, а первое число становится последним.

Каждую секунду будем выписывать первое число каждой последовательности в новый массив. Для всех чисел 1 до m найдем самый длинный **подотрезок** этого массива, все элементы которого равны этому числу.

Будем проделывать эту операцию на протяжении 10^{100} секунд. Для каждого числа от 1 до m определите самый длинный из подотрезков, найденных за это время.

Формат входных данных

В первой строке входных данных записаны два числа n и m ($1 \leq n, m \leq 100\,000$) — количество последовательностей и максимальное число, которое может встретиться в последовательностях.

В следующих n строках даны сами последовательности. В каждой строке сначала записано число k_i ($1 \leq k_i \leq 40$) — количество чисел в последовательности, а затем ещё k_i целых положительных чисел — сама последовательность. Гарантируется, что числа в каждой последовательности попарно различны и не превосходят m .

Суммарная длина всех последовательностей не превосходит 200 000.

Формат выходных данных

Выведите m чисел, i -е из которых равняется длине самого большого подотрезка, все числа в котором равны i и который встретился в выписываемом массиве за первые 10^{100} секунд.

Примеры

circular-cipher.in	circular-cipher.out
3 4	2
3 3 4 1	1
4 1 3 4 2	3
3 3 1 4	2
5 5	3
2 3 1	1
4 5 1 3 2	4
4 2 1 3 5	0
1 3	1
2 5 3	
4 6	0
3 4 5 3	0
2 6 3	2
2 3 6	1
3 3 6 5	1
	2